Name:

Honor Pledge (following http://www.rochester.edu/college/honesty/policy.html#pledge):

I affirm that I will not give or receive any unauthorized help on this exam, and that all work will be my own.

(signature/date)

problem 1
problem 2
problem 3
TOTAL

do not write on this page

1. (40 POINTS) We have n items arranged in a line. The i-th item has weight w_i and value v_i. Assume that the weights are positive integers and the values are positive reals. We also have a weight limit L (where L is an integer). We want to select a subset of the items that maximizes the total value and satisfies the following two conditions:

   - the total weight of the elements in the subset is at most L, and
   - no three consecutive items are selected.

   We will solve the problem using dynamic programming. Let $T[i, \ell]$ be the maximum total value of a subset of the first i items that satisfies the following two conditions:

   - the total weight of the elements in the subset is at most ℓ, and
   - no three consecutive items are selected.

   Give an expression (or a piece of code) to compute the value of $T[i, \ell]$ from smaller subproblems. **Clearly explain your expression** (for each formula explain what type of optimal solution does it correspond to).

2. (40 POINTS) We are given n numbers $a_1, \ldots, a_n$ and another number L. We want to find the length of the shortest increasing subsequence[1] of $a_1, \ldots, a_n$ whose sum is at least L. For example, if the input is $1, 2, 4, 10, 11, 7, 8, 10$ and $L = 28$ then the answer is 4 (e.g., take the subsequence $4, 7, 8, 10$).

   We are going to solve the problem using dynamic programming. Let $T[i, j]$ be the maximum sum of an increasing subsequence of $a_1, \ldots, a_i$ that ends with a_i and has length j (if $a_1, \ldots, a_i$ contains no increasing subsequence that ends with a_i and has length j we let $T[i, j] = -\infty$). (After the table is computed we will find the smallest j such that for some $i \in \{1, \ldots, n\}$ we have $T[i, j] \geq L$.)

   Give an expression (or a piece of code) to compute the value of $T[i, j]$ from smaller subproblems. **Clearly explain your expression**.

[1] A **subsequence** of a sequence $a_1, \ldots, a_n$ is a sequence $a_{i_1}, a_{i_2}, \ldots, a_{i_\ell}$ where $\ell \in \{0, \ldots, n\}$ and $1 \leq i_1 < i_2 < \cdots < i_\ell \leq n$.

3. (40 POINTS) A **shuffle** of two strings $A[1..n]$ and $B[1..m]$ is formed by interspersing the characters into a new string, keeping the characters of A and B in the same order (for example, '**s**ev**e**r**al**' is a shuffle of 'seal' and 'evr'). We are given 3 strings $A[1..n], B[1..m], C[1..p]$. We want to check whether there exists a shuffle of $A[1..n]$ and $B[1..m]$ that is a subsequence[2] of $C[1..p]$. We will solve the problem using dynamic programming. Let $T[i, j, k]$ = true if and only if there exists a shuffle of $A[1..i]$ and $B[1..j]$ that is a subsequence of $C[1..k]$. Give an expression (or a piece of code) to compute the value of $T[i, j, k]$ from smaller subproblems. **Clearly explain your expression**.

[2] A **subsequence** of a string $C[1..p] = C[1]C[2]\dots C[p]$ is a string $C[i_1]C[i_2]\dots, C[i_\ell]$ where $\ell \in \{0, \dots, p\}$ and $1 \le i_1 < i_2 < \dots < i_\ell \le p$.